

# TASK 3-Minikube Deployment Task

---

## Step 1: Start Minikube

Start the Minikube cluster using the following command:

minikube start

```
C:\Users\thami>minikube start
* minikube v1.35.0 on Microsoft Windows 11 Home Single Language 10.0.26100.3194 Build 26100.3194
* Automatically selected the docker driver. Other choices: virtualbox, ssh
* Using Docker Desktop driver with root privileges
* Starting "minikube" primary control-plane node in "minikube" cluster
* Pulling base image v0.0.46 ...
* Downloading Kubernetes v1.32.0 preload ...
  > preloaded-images-k8s-v18-v1...: 333.57 MiB / 333.57 MiB 100.00% 487.11
  > gcr.io/k8s-minikube/kicbase...: 500.31 MiB / 500.31 MiB 100.00% 660.36
* Creating docker container (CPUs=2, Memory=4000MB) ...
! Failing to connect to https://registry.k8s.io/ from inside the minikube container
* To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
* Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
  - Generating certificates and keys ...
  - Booting up control plane ...
  - Configuring RBAC rules ...
* Configuring bridge CNI (Container Networking Interface) ...
* Verifying Kubernetes components...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Enabled addons: storage-provisioner, default-storageclass
```

This initializes the Minikube cluster using Docker as the driver.

## Step 2: Install Kubectl

Since Kubectl is not found, install it with the following command:

sudo snap install kubectl --classic

Alternatively, you can download it using curl:

```
100 138 100 138 0 0 216 0 --:--:-- --:--:-- --:--:-- 215
100 64 100 64 0 0 51 0 0:00:01 0:00:01 --:--:-- 524
k8s.io/kubectl v1.32.0
```

```
curl -LO "https://dl.k8s.io/release/${curl -L -s
https://dl.k8s.io/release/stable.txt}/bin/linux/amd64/kubectl"
sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
```

## Step 3: Verify Kubectl Installation

Check the client version to confirm successful installation:

kubectl version --client

```
$ sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
$ kubectl version --client
Client Version: v1.32.3
Server Version: v5.5.0
```

## Step 4: Create a Deployment

Create a deployment named `pod1` with the image `shankar4112/devops-training`:

```
kubectl create deployment pod1 --image=shankar4112/devops-training --port=80
```

```
C:\Users\thami>kubectl create deployment r1 --image=thamilvasanth/devops --port=80
deployment.apps/r1 created
```

## Step 5: Expose the Deployment

Expose the deployment as a NodePort service:

```
kubectl expose deployment pod1 --port=80 --type=NodePort
```

```
C:\Users\thami>kubectl expose deployment r1 --port=80 --type=NodePort
service/r1 exposed
```

## Step 6: Verify the Pod

Check the running pods:

```
kubectl get pods
```

```
C:\Users\thami>kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
r1-7768f67d8d-r2vqk               1/1     Running   0           35s
```

## Step 7: Access the Service

Expose the service using Minikube and get the URL:

```
minikube service pod1
```

```
C:\Users\thami>minikube service r1
```

NAMESPACE	NAME	TARGET PORT	URL
default	r1	80	http://192.168.49.2:32077

```
* Starting tunnel for service r1.
```

NAMESPACE	NAME	TARGET PORT	URL
default	r1		http://127.0.0.1:57672

```
* Opening service default/r1 in default browser...  
! Because you are using a Docker driver on windows, the terminal needs to be open to run it.
```

## Step 8: Output in the Web Browser

